Extraction of Computational Models of Narrative from Natural Language

Josep Valls-Vargas
Computer Science Department
Drexel University
josep.vallsvargas@drexel.edu

Friday 6$^{\text{th}}$ June, 2014

Abstract

Computational narrative focuses on methods to algorithmically analyze, model, and most importantly, generate narratives. Most current work requires the use of handcrafted narrative models in some knowledge representation formalism, which are expensive to generate. This document focuses on exploiting natural language processing (NLP) techniques to acquire those models automatically from natural language text. Specifically, we first overview existing work on formal models of narrative, and then we present an overview of natural language processing techniques for extracting structured information from text. After that, we review work that bridges the gap from NLP to computational narrative by allowing computational narrative methods to feed directly from narratives written in natural language.

Contents

1 Introduction

Computational narrative is an emergent field of research at the intersection of traditional narratology, artificial intelligence, natural language processing and cognitive science. The field is still in an early stage of defining itself [RFW09] but computational approaches for narrative modeling have seen some increased interest in recent years with conference workshops like *Computational Models of Narrative*[1], or *Intelligent Narrative Technologies*[2].

Narratives are found to some degree in all kinds of everyday situations. From news reports in newspapers or magazines, to scientific articles and informal conversations with peers and friends. Narratives are therefore a key element in human communication [Cha80]. Moreover, computational models of narrative have applications to tasks such as machine translation, automatic summarization, information extraction, training, education, or entertainment. For example *Narrative Science*[3] is a reporting services company that has been in the spotlight lately as they have been providing automatically generated written reports for Forbes magazine[4].

A notable problem is that most of the systems currently used rely on handcrafted narrative models in some knowledge representation formalism for their computational models of narrative. In the case of text generation, systems like the ones used by Narrative Science require hundreds or thousands of human authored templates. Moreover, the models required tend to use complex notation and need to be authored by knowledgeable technical people. This represents a heavy burden as the models or templates for one system typically cannot be reused by another, leading to an authorial bottleneck [Ger09, Bat97].

However, there is a vast amount of material in the form of natural language that is currently available. The existing body of literature could be exploited by those systems if we could enable automatic extraction of the required models from text. An added benefit would mean allowing non-technical people to feed knowledge into those systems by using natural language. This document focuses on exploiting natural language processing (NLP) to automatically generate narrative models from natural language text.

This document is structured as follows: First, Section 2 gives an introduction on narratology and presents computational approaches to modeling narratives and some applications. In Section 3 we overview the state-of-the-art of natural language processing techniques and available tools focusing on the tasks relevant to the extraction of structured information from text. Then, in Section 4 we survey some comprehensive approaches that attempt extracting and processing narrative information from natural language text. Finally, Section 5 discusses some of the open research questions and presents our current work and future work plan for extracting narrative information from text.

2 Computational Narrative

Narratives are found to some degree in all kinds of everyday situations. From news reports in newspapers or magazines, to scientific articles and informal conversations with peers and friends. Narratives are therefore a key element in human communication [Cha80]. Because of the importance of narrative, and with the

[1] http://narrative.csail.mit.edu/cmn14/
[2] http://int7.westphal.drexel.edu/
[3] http://narrativescience.com/
[4] http://www.forbes.com/sites/narrativescience/

$$
\text{Narrative}\left\{\begin{array}{l}
\text{Story}\left\{\begin{array}{l}
\text{Events}\left\{\begin{array}{l}\text{Actions}\\ \text{Happenings}\end{array}\right.\\
\text{Existents}\left\{\begin{array}{l}\text{Characters}\\ \text{Settings}\end{array}\right.
\end{array}\right.\\
\text{Discourse}\left\{\begin{array}{l}
\text{Structure of narrative transmission}\\
\text{Manifestation}\left\{\begin{array}{l}\text{Verbal}\\ \text{Cinematic}\\ \text{Pantomimic}\\ \text{etc.}\end{array}\right.
\end{array}\right.
\end{array}\right.
$$

Figure 1: Chatman's taxonomy of narrative elements. In Chatman's work, plot is also referred as *story* or *content*; discourse is also referred as *expression*.

popularization of technology that enables computers to process large amounts of data, the field of computational narrative arose with the goal of algorithmically analyzing, modeling, and automatically generating narratives.

A narrative is an account of connected events, real or imaginary, that forms a story. Although narratives can be found in many forms and media in this document we will focus on written stories. We will reference the audience of the narrative as the "reader", but most concepts also apply to an audience watching a movie or listening to a live storyteller.

Over the years, there has been considerable debate about the extent of what can be considered as narrative [Rya87, Man12]. It is commonly accepted though that narratives must be built from more than one event and that those events should be connected to one another as a coherent whole [Man12]. Literary scholars such as Scholes and Kellogg [SPK06] studied the history of storytelling and differentiate two varieties of narrative: 1) *empirical narratives that involve allegiance to reality, emphasizing truth*, and 2) *fictional narratives that involve allegiance to an ideal, emphasizing aesthetic notions such as beauty and moral ones such as goodness*. Both share some common structural elements across different media. These structural elements include representation of different phenomena that are relevant to making sense of a narrative, such as plot (or story), discourse (or expression), time, point-of-view (or focalization) and embedded narratives.

In the remainder of this section we will first introduce the relevant narrative concepts of plot and discourse (Section 2.1). We will then present an overview of narratology theories about those narrative concepts that have constituted the foundation of most computational narrative work (Section 2.2). After that, we will move on to present an overview of computational approaches to model narrative elements and how to annotate written text with narrative information (Section 2.3). Finally we will present some applications of computational narrative to digital entertainment (Section 2.4).

2.1 Plot and Discourse

Stories can be told in many different ways. Therefore, narratology theory not only looks at the stories themselves but also at how those stories are told. Chatman [Cha80] presented a unified study of narrative, distinguishing stories themselves (plot) from they way they are told (discourse) and provided rigorous definitions and systematic analysis methods for narrative theory. Chatman [Cha80] presented a unified study of narrative, defining it as a broad form of semiotic communication. Chatman argues that the study of narrative should concentrate on underlying inter-medial patterns and elements rather than medium-specific micro-elements. Chatman's work defines a dual model that separates the plot (i.e., what is being told) and the discourse (i.e., how it's being told). Furthermore, Chatman classified characteristic elements from the plot of a narrative and the means in which the narrative is communicated (Figure 1).

Although the plot constitutes the core of a story, it is the discourse that may have the greatest impact on how a narrative is perceived by the audience. A typical example is the movie *Rashomon* (Akira Kurosawa, 1950) where the story of a crime is recalled from the point-of-view of different characters, each contributing to a different narrative even though they are all talking about the same story (or plot). Different resources and limitations across different media may be another example of how a story told using oral storytelling can vary greatly if retold using motion pictures. Focusing on the written medium, a story can be told from a neutral objective perspective or from a number of subjective (and potentially biased) perspectives or points of view (called focalization). While news reports should try to be written in an objective and informative manner, a narrator in a work of fiction will very frequently use discourse resources to convey a specific authorial goal or appeal to the reader's sensibility and emotions.

The discourse of a narrative also defines the amount of information regarding the plot that is deemed implicit or left to the reader's interpretation. Schank and Abelson [SA77], in their work, pointed out that many causal relations in natural language text are not explicit and instead depend on inferred intermediate events. A widely used example that illustrates this point, sometimes attributed to Ernest Hemingway, is the 6-word story:

> For sale: baby shoes, never worn.

The discourse of this short story leaves a lot for the reader to interpret. For example, one reader could interpret that the story refers to a miscarriage. But another reader might interpret that the pregnancy ended up with a baby of the opposite gender than expected. The discourse might be ambiguous about the specific details of the plot and the reader's assumptions may not be precise.

2.2 Narratology

Narratology is a discipline dedicated to the study of the logic, principles, and practices of narrative representation and storytelling, including common themes, conventions and symbols. Core elements and ideas for narratological modeling of narrative were introduced by Greek philosophers. Narratology has been dominated by structuralist approaches since the 1900s, and has developed into a variety of theories, concepts, and analytic procedures [Man12].

In the following subsections we will look at some narratology theories from which ideas for computational models of narrative have been borrowed. We will present thematic narratology, tightly coupled with the study of the plot of a narrative; modal narratology that studies the discourse; and, finally; cognitive narratology that focuses on the reader's interpretation of a story.

2.2.1 Thematic Narratology

Thematic narratology focuses on the semiotic formalization of the sequences of the actions told in the narrative. One of the main influences was the structuralist work of Propp [Pro73]. Propp focused on an empirical analysis of Russian folk tales and presented a model of the elementary components of narratives and the way they are combined. Propp describes a particular type of Russian fairy tales in terms of a sequence of thirty-one abstract "functions" (such as, *abstention*, encoded by the symbol β or *struggle*, encoded as H) that are in turn broken into subfunctions (such as, β^2, death of parents; β^3, departure of siblings; H^1, fight in open field; or H^2, contest). The functions are, in turn, grouped in sequences forming narrative episodes called "moves" and multiple moves can be found in a single story. Propp also postulated that the characters that appear in the stories could be categorized into 7 character functions or roles (such as, hero or villain). Often, characters can play more than one role or there can be more than one character performing the functions for a given role.

Propp's functional model served as a reference for the elaboration of story grammars [SLTW05] and it has been used as the basis for several computational models described below [GDAPH05, Fai07], including some of our previous and current work [VVOZ13a, VVOZ14]. In Propp's work, the discourse is often neglected and his analysis is based on an omniscient knowledge of the plot of the stories. As evidence of this, in his work [Pro73], Propp considered events that are not explicitly stated in the stories but implicit or left for the reader to interpret.

Later, in Section 2.3.2 we will present computational models for plot inspired by Propp's work and Labov's minimal model of a story defined as a series of states and transitions between states [Lab72, Ger09].

2.2.2 Modal Narratology

Modal narratology [GL83] focuses on the way a story is told during a storytelling session, stressing voice, point of view, transformation of the chronological order, rhythm and frequency. Modal narratology is based on pre-structuralist theories of narrative and analyzes perspective, time, logic and rhetoric. For example, modal narratology explores the idea of *focalization* [Ger09], the narrator's temporal and cognitive position regarding the characters, and introduces the fundamental distinction between "narrated time" and "time of narration". In this line of work, a more complex model of the plot is needed since it needs to encode the perspectives of different characters and causal event information that does not unfold until the author explicitly tells the reader. Modal narratology aligns closely with the discourse model that can be directly extracted from the text. From a computational perspective this model poses several issues because of the ambiguous nature of natural language, yet there have been annotation approaches that seek to symbolically encode modal aspects of a narrative [Man12].

Later, in Section 2.3.1 we will present computational models for discourse that draw from some ideas from modal narratology.

2.2.3 Cognitive Narratology

Other theories exist that attempt to model narratives from different perspectives. Some of the later trends borrow from cognitive science to model not only stories but also how a reader perceives and internalizes the development of a narrative. Cognitive narratology [Her00] focuses on the human intellectual and emotional processing of narratives and looks at the interpretation of the character's purposes and motivations in order to infer specific narratology primitives like volitional cause and logical sequences of actions.

This approach is not restricted to literary narratives. Everyday oral narratives are considered to represent an underlying anthropological competence in its original form [Flu96]. Cognitive approaches have influenced AI research targeted at modeling and simulating human narrative intelligence [Her00]. From those, the term "hermeneutics" has been borrowed to describe computational approaches to story understanding [MS99b].

2.3 Computational Models of Narrative

Artificial intelligence has tried to model narratives and understand human cognition since the 1970's [MS99b]. The interest in computationally modeling narratives has received an increased interest in recent years. This interest can be seen in recent projects like DARPA's *Narrative Networks* (N2) or conference workshops like *Computational Models of Narrative*[5].

The remainder of this section presents an overview of the existing work on computational models of narrative. The models can be grouped in different levels or layers. We use two broad categories closely related to the distinction between plot and discourse (illustrated in Figure 2): a) modeling discourse structures, in terms of functional organization (e.g., elaboration, comparison, etc.) and narrative primitives (e.g., motivation, cause, etc.); and; b) modeling plot at a conceptual level, independent from text, in terms of plot elements (i.e., Chatman's taxonomy) or states and transitions between states (i.e., plot points).

2.3.1 Computational Models for Discourse

The discourse structure of a text defines a logical flow of events, states, and propositions that result in a coherent exposition of an idea, argument, or story. This structure is defined as a set of relations between text spans. Some of the relations overlap with narrative primitives from the cognitive literature (e.g., volitional cause), whereas other relations are independent (e.g., elaboration, concession, etc.). In this section we present different models for discourse operating at different conceptual levels.

A widely known approach to text discourse annotation is the *Penn Discourse Treebank* (PDTB) [PDL$^+$08] project. The goal of the project is to annotate the million-word *Wall Street Journal* corpus in the *Penn TreeBank* [MMS93] with a layer of discourse annotation. The PDTB defines several types of discourse relations. The relations bind together *entities* ("existents" in Chatman's taxonomy, see 1), *events* ("actions" and "happenings" in Chatman's taxonomy) or *propositions* (assertions, beliefs or intentions of the characters). There are different types of discourse relations that account for different phenomena in the discourse of a text,

[5] `http://narrative.csail.mit.edu/cmn14/`

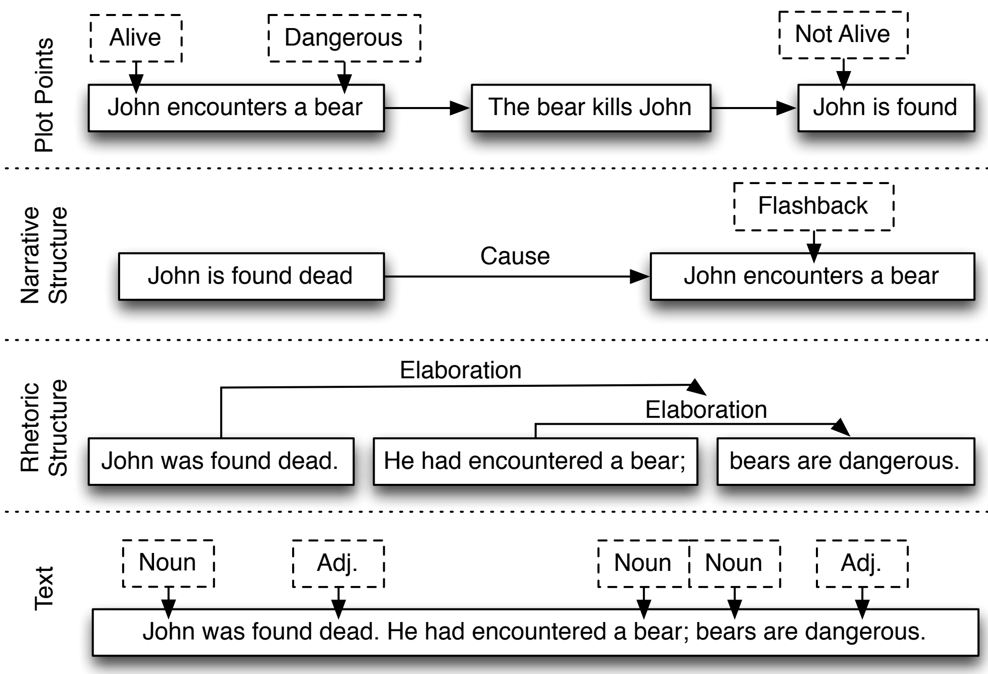

Figure 2: Different layers of abstraction from a text and exemplified models for each each layer. The solid arrows indicate relationships between elements and the dashed boxes properties.

the main groups being temporal, contingency, comparison, or expansion. The relations are called *discourse connectives* and the events or propositions they refer to are called their *arguments*. The discourse connectives can be explicit or implicit, have additional attributes and its arguments can be continuous, multi-clausal or non-clausal. The PDTB discourse relation annotation scheme is designed to annotate discourse connectives and their arguments, as well as some additional information regarding their meaning and attribution.

The *Rhetorical Structure Theory* (RST) [MT88], is another annotation scheme to annotate discourse relations and their arguments. The relations are dealt in terms of subordinating conjunctions (when, because...), coordinating conjunctions (and, but...) and other structural relations. RST divides the text into non-overlapping chunks called *elementary discourse units* (EDU). Consecutive EDUs are related to each other using a set of discourse relations forming a tree structure. In each relation, each EDU plays the role of the *nucleus* or the *satellite* of the relation. Several types of relations are defined, such as "enablement" or "means".

From a computational perspective, the PDTB dataset has been used in computational experiments for validating discourse structure theories [PDL+08]. There have also been some attempts to automatically extract discourse relations. For example, HILDA [HPI10] is a discourse parser that uses *Support Vector Machines* (SVM) to learn how to produce RST parse trees.

The discourse models described so far deal with structural elements of the text rather than higher-level narrative primitives.

Sequence is a very important aspect of a narrative closely related to time. A narrator may decide to tell the events of a story in a non-chronological order, and thus time anchors are a key factor in reconstructing the plot from the discourse. Time references are embedded in natural language and modeling time in natural language text is relevant to many areas of research such as question answering [PCI+03]. *TimeML* [PCI+03] is a widely used markup language for annotating events, temporal expressions and time links in natural language text. An event is a core element of the plot of a narrative (see Figure 1). Events may be complex elements that can refer to a punctual point in time or can last for a period of time. *Temporal expressions* are sequences of words that tell when something happened, how long something lasted, or how often something occurs. At the same time, temporal expressions may be calendar dates, times of day, or durations, such as periods of hours, days, or years. Finally, *time links* are relationships between times, events, or events and times. They indicate particular relationships like events that happen at the same time or events happening for the duration of one another. TimeML uses Allen-style interval relations [All84] to encode time links. TimeML is tightly coupled with elements found at a low level like verbs tenses, adverbs or adjectives. Because of the nature of natural language, it's worth mentioning that temporal expressions can be precise or ambiguous.

There has been some research in connecting discourse relations with higher-level structures of narrative discourse in order to infer elements like purpose or motivation. For example, *Temporal Discourse Models* (TDMs) [MP04] are tree-structured syntactic models of discourse structure that combine a discourse tree with temporal constraints (i.e. partial orderings) on the events described in the tree.

Moving to an even higher level of abstraction, there are several other approaches and proposed annotation schemes with the goal of automatically understanding and reasoning about narratives. *NarrativeML* [Man12] is a markup language that seeks to fully model the discourse of a story including several narrative primitives and information about the narrator's point-of-view (i.e., focalization), reader's mental model, embedded narratives, character affect states, beliefs, desires and goals.

2.3.2 Computational Models for Plot

Some applications require the plot to be conceptually abstracted from the discourse of the text. Here we present how different areas of research have approached the modeling of plot using different formal representations.

Ontologies and taxonomies can be used to classify topics, plot structures or plot elements. Different classification schemes have been used to classify elements using different theories. For example, Chatman's taxonomy differentiates elements such as events (actions or happenings) from entities (existents: characters, settings, props, etc.). Furthermore, theoretical structuralist work resemble formal languages and their approaches translate well into formal logic systems such as *grammars*. The plot of a story can be represented by using grammars with symbols representing key events (or plot points). Proppian functions can be used for classifying recurrent plot structures whereas Propp's hierarchy of moves and character functions has been used to define grammars. Both approaches have been used in different generation systems [SLTW05, GDAPH05].

Considering a minimal model of the plot of a story as a chronologically ordered set of plot points, we can see how ideas from the field of automated *planning*, can be used to model plot. Events in a story can be defined as planning operators and its preconditions and postconditions for defining the temporal and causal links in the plot of the story. The planning operators can be defined to closely match the text by aligning them with verb frames or can be matched with higher-level elements in the plot with an abstracted and more conceptual interpretation. Recently, at the Intelligent Narratives Workshop 6 (2013), the Narrative Formalization Using PDDL (Planning Domain Definition Language) panel explored using PDDL to encode the plot of The Iliad. Systems that use planning often rely on extensions of PDDL that encode additional information on top of the planning operators. *Tale-Spin* [Mee76] is an early example that uses a planning system to generate the plot of a story and a text realization component.More recently, planning approaches have been used to model stories and encode authorial intent using partial-order planning in the context of story generation, improving over the results achieved by Tale-Spin [RTB11]. In our previous work, we used planning approaches to infer spatial constraints about locations in a given story and used those later to generate a map representation of the virtual environment where the story unfolds (with applications to automatic game design) [VVOZ13b].

Scripts are a method of representing chains of events or episodic knowledge, an example of which is the sequence of events in the plot of a story. Scripts in the context of AI were introduced by the work of Schank and Abelson [SA13] as a formalism drawing from cognitive science as an attempt to natural language understanding. As in planning, different levels of conceptual abstraction can be used. Verb frames can be seen as events and the sequence of verb frames forming a script is then a plot structure [CJ09]. Scripts can also represent recurring patterns or conceptual narrative plot structures [MS99b].

2.3.3 Annotating Narratives

Several annotation schemes and annotation environments have been proposed that combine layers of discourse and plot annotations on top of natural language text. These annotation schemes add different layers of computer readable information representing different parts of the narrative. The annotations can then be

Figure 3: An excerpt of an example story generated by *Tale-spin*.

> Joe Bear was hungry. He asked Irving Bird where some honey was. Irving refused to tell him so Joe offered to bring him a worm if he'd tell him where some honey was. Irving agreed. But Joe didn't know where any worms were, so he asked Irving, who refused to say.

used to extract different models of the plot or discourse of the narrative for further processing. For example a model of a sequence of states from a story can be extracted in the form of a script-like representation for plot points [Fin09].

The *Proppian fairy tale Markup Language* (PftML) project [Mal01][6] is an annotation scheme based on XML that implements a DTD (Document Type Definition) to standardize a formal analytical model for tales based on Propp's work. This annotation specification provides a computer readable format for the plot of the narrative using Propp's functions. The *Story Workbench* project [Fin11][7] provides an integrated framework supporting several annotation layers. It provides a number of common text annotation operations, including representations (e.g., tokens, sentences, parts of speech), functions (e.g., generation of initial annotations by algorithm, checking annotation validity by rule, fully manual manipulation of annotations) and tools (e.g., distributing texts to annotators via version control, merging doubly-annotated texts into a single file). It was designed to support multiple annotation workflows and it integrates with version control systems. The Story Workbench has been used to annotate a dataset of 15 stories with NLP information (discourse) and Proppian functions and characters (plot). A related but different approach is the Scheherazade system [EM07][8]. The Scheherazade system is a platform for symbolic narrative annotation. It provides tools to define common sense knowledge regarding a story and then a graphical interface for manually encoding narrative semantics such as timelines, states, events, characters and goals. Although narrative primitives can be mapped to text, the system can be used for generating symbolic representations of a story independent from the text.

2.4 Applications

Computational models of narrative have many applications. In this section we focus on applications from the digital entertainment research community that use some of the models described previously.

2.4.1 Computational Creativity and Storytelling

As we have already seen, narratives are complex intellectual products at the plot level. Additionally, storytelling (at the discourse level) is an activity that requires a wide range of skills and cognitive abilities intrinsic to humans. Despite that, there have been some research efforts aiming at achieving computers inventing and telling stories. In the context of storytelling, creativity implies inventing a satisfactory story in terms of believable characters, their personalities, goals, feelings and emotions; interesting situations and events; and a discourse that facilitates the reader to understand the characters' motivations, associate characters intentions with feelings, and, develop empathy towards the characters and situations [Ger09].

Tale-spin [Mee76] is a story generation system that generates stories of woodland creatures using backward chaining for resolving goals and subgoals and forward chaining to compute consequences of narrative events. Tale-spin simulates a small world populated with characters, each with their own problems and motivations. Each character uses planning in order to satisfy their own goals given the current world state. Complex relations between characters are modeled using simulation over the events and the outcome is used to select plot points with satisfied preconditions. After the simulation and problem-solving phase, a separate component is used to generate a textual description of the generated narrative. Figure 3 shows an excerpt of an example story generated by *Tale-spin*.

[6]Available: `http://clover.slavic.pitt.edu/sam/propp/theory/propp.html`
[7]Available: `http://projects.csail.mit.edu/workbench/`
[8]Available: `http://www.cs.columbia.edu/~delson/software.shtml`

Minstrel [Tur93] tells stories about the Knights of the Round Table. Minstrel uses case-based reasoning instead of planning, and introduces concepts such as author goals in order to guide the plot generation. Minstrel implemented a general theory of creativity based around the concept of Transform Recall Adapt Methods (TRAMs). TRAMs are used to query an episodic memory and retrieve schemas or scripts that define the story. In the case that a query fails, the query process allows TRAMs to be modified and additional queries issued recursively, the output of which can be combined and adapted to match the initial query constraints. *ProtoPropp* [GDAPH05] generates stories step by step by using a simplified Case-Based Reasoning approach with a Proppian ontology that transforms annotated stories using a simulator and explicit domain knowledge.

2.4.2 Player Engagement in Computer Games

Storytelling not only happens in text but in other media such as computer games. Computational models of narrative have been used to achieve systems that automatically adapt the plot of a game to maximize engagement of the player at hand.

For example, *OPIATE* [FC03, Fai07] is an interactive storytelling engine that generates new stories using a Case-Based Reasoning (CBR) approach that reuses tales analyzed in terms of Proppian functions. The *Automated Story Director* [RSDA08] uses a planning approach to model a narrative space. It automatically considers all possible breaks in the original story that can be caused by the actions of the player and proposes contingency narratives for each rupture, in order to allow the game narrative to proceed. *PAST* [RB12] builds on top of the *Automated Story Director* and uses it in combination with player modeling to generate player-specific game narratives while still maintaining authorial intent on the original narrative space.

2.4.3 Procedural Content Generation for Computer Games

Procedural content generation (PCG) refers to the creation of content automatically through algorithmic means [TYSB10]. PCG has been applied to map generation for computer games for some time in order to increase the variability and replayability of games [DP11]. Some computer game genres require meaningful stories and complex worlds in order to successfully engage players.

Recent work has started to integrate narrative components into map generation. *Game Forge* [HZDR11], is a system capable of generating a map given a linear story represented by a sequence of plot points. It uses a genetic algorithm approach to infer the spatial relationships between the locations annotated in the given story and generates a map genotype as a space tree. In a second step, the system maps the genotype to a phenotype where the space tree has been embedded on a grid. Then, the virtual world is graphically realized as a 2D map by instantiating predefined image tiles and handed to a game engine so that it can be navigated by the player's avatar. In our own research [VVOZ13b], we proposed a framework which, given the specification of a *story space*, represented as a collection of *plot points* and their dependencies, can generate maps that support one or more stories from that story space. Our system searches in the space of possible spatial configurations of the map, determining the set of stories that can unfold in each of those configurations. Then, using automatic story evaluation techniques, it determines the narrative quality of each possible story, the combination of which determines the overall quality of the map. Finally, the map is *realized* into a graphical representation.

3 Natural Language Processing

In this section we present an overview of the state of the art in Natural Language Processing (NLP) techniques, specifically focused on those that can be used to automatically generate computational models of narratives from natural language text.

In the remainder of this section, we will first describe the different subfields and tasks involved in the NLP pipeline, and how they relate to each other (Section 3.1). Then we will present work on information extraction from text (Section 3.2) and how to complement it with common sense knowledge (Section 3.3).

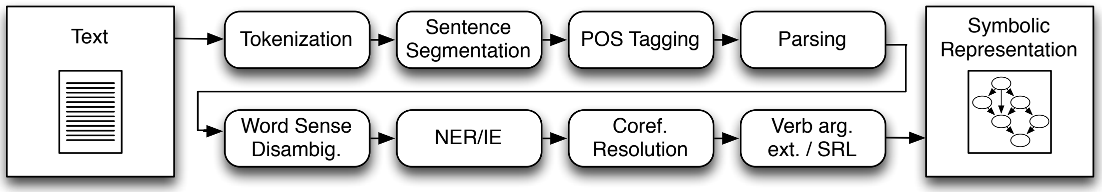

Figure 4: Typical NLP pipeline, specially prevalent for applications related to text understanding.

3.1 The Natural Language Processing Pipeline

The field of Natural Language Processing (NLP) has received a lot of attention since the early days of computing and artificial intelligence. NLP has had successful applications in the areas such as Information Extraction (IE) or Machine Translation (MT). The history of NLP generally starts in the 1950s. Some notably successful NLP systems were developed in the 1960s but up to the 1980s, most NLP systems were based on complex sets of hand-written linguistic rules. Starting in the late 1980s, however, there was a revolution in NLP with the introduction of machine learning algorithms for natural language processing [MS99a]. The latest trends in NLP involve using statistical methods and machine learning algorithms and the increase in computational power and data availability (i.e. *big data*) has led to a renewed interest in NLP research [MJ12].

NLP is a broad research area addressing a range of different problems. Some of the most common NLP tasks are:

- **Information Extraction (IE)**: focuses on automatically extracting structured information from unstructured textual documents. IE plays a big role in document classification, indexing and retrieval [CL96].
- **Text Understanding**: also known as machine reading, is related to IE, but differs in that its scope goes beyond information storage and seeks to extract a structured representation that enables further computation, such as problem solving applications to operate on natural language input [Bob64].
- **Text Generation**: also known as natural language generation (NLG) studies methods to convert computational representations of information into readable human language. Dialogue agents or interactive fiction systems use NLG to communicate an internal state to the user. Paraphrasing and summarization are NLG tasks heavily related to text understanding. NLG is sometimes considered a specific case of Procedural Content Generation (PCG) [VVOn12].
- **Spell and Grammar Checking**: studies automatic methods to verify spelling and grammatical correctness of written natural language. Text is parsed and compared against language models and dictionaries in order to identify possible errors. Checking is related to predictive text systems and automatically correcting output from speech recognition or optical character recognition (OCR) systems [MJ12].

Work on addressing those problems shares a collection of tasks and subtasks (such as tokenization or grammatical analysis). Because there is an intrinsic interdependency between some of those common tasks, a pattern for a pipeline has emerged combining some tasks required for later text processing stages. Figure 4 shows a workflow diagram for this pipeline [JBDN05, CRHS10]. Some of the most relevant tasks for our focus area are:

- **Segmentation, Chunking and Tokenization**: given a string of characters representing a chunk of text (or document), separate words, sentences or other elements like paragraphs, symbols or individual morphemes from compound words.
- **Part-Of-Speech (POS) tagging**: given a sentence, determine the part-of-speech each word plays for a specific language grammar (e.g. “the red truck” $\Rightarrow$ the $\mapsto$ *determiner*, red $\mapsto$ *adjective*, truck $\mapsto$ *noun*).

- **Morphological Analysis**: given one or more tokens, identify individual morphemes and recognize the lemma, stem or root for inflected words along with inflection information (e.g. "singing" ⇒ sing ↦ *lemma*, -ing ↦ *present continuous*).
- **Grammatical and Syntactic Parsing**: given a sentence, generate the structures of the syntactic and grammatical roles and relationships between the words and phrases (e.g. "My truck is red." ⇒ *(S (NP (PRP$ My) (NN truck)) (VP (VBZ is) (ADJP (JJ red))) (. .))*).
- **Entity Extraction and Named Entity Recognition (NER)**: given a document, identify items in the text that map to specific entities and recognize the type of each such entity (e.g. "Barack Obama is the president of the United States" ⇒ Barack Obama ↦ *person*, United States ↦ *country*).
- **Coreference Resolution**: given a document, determine which words (mentions) refer to the same entities (e.g. "Alice is a beautiful girl. She is also intelligent." ⇒ *She* ↦ *Alice*).
- **Attribute and Relationship Extraction**: given a document, identify the relationships between entities and their attributes. (e.g. "An old man lived with his daughter and younger son." ⇒ *is(man, old); has(man, daughter); has(man, son); is(son, young)*.
- **Word Sense Disambiguation (WSD)**: given a token with multiple possible senses, identify the intended sense in the context it appears (e.g. "He was sent to the pen for burglary." ⇒ *pen* ↦ *penitentiary*).
- **Semantic Role Labeling (SRL)**: given a verb with multiple arguments (i.e. subject, direct object and multiple indirect objects), map the arguments to semantic roles (e.g. "Bob bought a computer from Alice." ⇒ *Bob* ↦ *buyer*, *Alice* ↦ *seller*, *computer* ↦ *good being sold*).
- **Discourse Analysis**: given a document, identify elements like the discourse structure of connected text or recognize and classify speech acts. For example, identify discourse connectives and their arguments [PDL+08] or identify relations between paragraphs in a document using rhetorical structure theory (RST) [HPI10].
- **Topic Segmentation, Recognition and Analysis**: given a document identify segments devoted to specific topics or identify the main topic, mood and sentiment. (i.e. "this movie sucks" ⇒ Sentiment ↦ *negative*).

While some of these tasks are trivial (e.g. stemming for English words), some of them may be much more complex (e.g. stemming heavily inflected morphology for German words), and some others (such as coreference resolution) are still open problems [PRM+11]. Focusing on the English language, tasks like POS tagging and Named Entity Recognition are mostly solved whereas some tasks like parsing or coreference resolution have shown good progress over the recent years. The task of word sense disambiguation has also received attention and improvements and the access to large datasets and crowdsourcing has greatly improved the tasks related to machine translation. When it comes to text classification, Spam detection is also mostly solved and tasks like sentiment analysis are receiving a lot of attention and making important steps forward. Other tasks that depend on deep understanding of the text like question answering, dialog generation, paraphrasing or automatic summarization still pose important unresolved problems. Table 5 summarizes the current progress of some NLP tasks for the English language.

Several projects and packages that implement some of those tasks have been released for public use for commercial and research purposes and have popularized and enabled further development of the NLP pipeline. For example, *Stanford CoreNLP* is a package that includes Java implementations of several Stanford NLP tools including the POS tagger, the named entity recognizer (NER), the Stanford parser, the coreference resolution system, and the sentiment analysis tools, and provides model files for processing English text [9]. *Natural Language Tool Kit (NLTK)* is a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning written in Python with interfaces to external tools and corpora. A book is also provided to introduce technical and non-technical people to developing

[9] Available: `http://nlp.stanford.edu/software/corenlp.shtml`

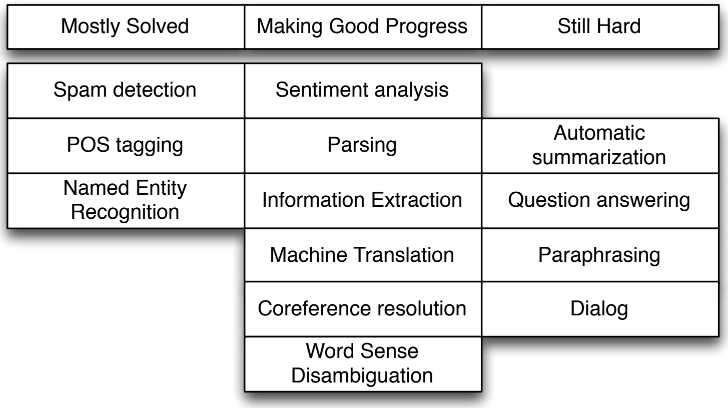

Figure 5: Progress in several tasks related to NLP. Adapted from slides from NLP MOOC by D. Jurafsky and C. D. Manning. Retrieved 2012: `https://www.coursera.org/course/nlp`

applications that feature NLP and computational linguistics [10]. *Apache OpenNLP* is another Java toolkit for the processing of natural language text supporting the most common NLP tasks, such as tokenization, sentence segmentation, POS tagging, named entity extraction, chunking, parsing, and coreference resolution. Also includes maximum entropy and perceptron based machine learning algorithms and models [11]. *General Architecture for Text Engineering (GATE)* is an integrated development environment for building NLP applications. It provides a user interface for several text processing workflows (including annotation and visualization), and includes ANNIE (A Nearly-New Information Extraction System) focused on IE tasks on top of many common NLP tasks and can be used programmatically using its Java API [12]. *The Curator (U. of Illinois)* is a Java suite integrating several natural language processing tools including a semantic role labeling (SRL) tool using verb frame information [13].

The following subsections elaborate on some of the NLP tasks that are particularly relevant to narrative information extraction from text.

3.1.1 Grammatical and Syntactic Parsing

The task of syntactic parsing tries to identify the syntactic roles and relationships between words and build a parse tree of the syntactic structure of a sentence. Such structures, along with the POS tags, are commonly referred to as layers of annotation.

Approaches to syntactic parsing typically combine formal grammars with statistical models to deal with the complexity of natural language (the use of simple grammars such as context free grammars is not feasible due to the ambiguity, complex syntax and size of the lexicon of any typical natural language). A common approach uses Probabilistic Context-Free Grammars (PCFG), which extend context-free grammars by assigning probabilities to production rules. The probability of a derivation (parse) is the product of the probabilities of the productions used in that derivation. A trained model including the production rules and the probabilities is typically computed by machine learning algorithms operating on a large corpus. Lexicalized PCFGs (where head words annotate phrasal nodes) are used for high performance PCFG parsing. Great success in terms of parse disambiguation and language modeling has been achieved by various lexicalized PCFG models [KM03]. More recent work has combined lexicalized and unlexicalized PCFG for phrase structure and lexical dependency parsing [KM03].

Statistical models for parsing, such as PCFGs are automatically trained using annotated corpora. Annotated corpora usually contain text and some layers of annotation, such as syntactic trees. The *Brown University Standard Corpus of Present-Day American English* was a pioneering project in the 1960' that collected and annotated a large corpus of literature for computational analysis. The project annotated POS

[10] Available: `http://www.nltk.org/`
[11] Available: `http://opennlp.apache.org/`
[12] Available: `http://gate.ac.uk/`
[13] Available: `http://cogcomp.cs.illinois.edu/page/software_view/Curator`

tags for over a million words. Nowadays there are several sources of annotated corpora with additional layers of annotation.A corpus annotated with syntactic trees is referred as a treebank. Some relevant treebanks are the *Penn Treebank Project*[14] and the *Penn Discourse Treebank Project*[15] from the University of Pennsylvania and distributed by the Linguistic Data Consortium (LDC). There are also domain-specific treebanks, for example the *GENIA treebank*[16], using the same annotation format but for the biomedical domain.

Concerning available software packages, the *Stanford Parser* provides Java implementations of probabilistic syntactic parsers including optimized unlexicalized and lexicalized PCFG and lexicalized dependency parsers. The lexicalized probabilistic parser implements a factored product model with separate PCFG phrase structure and lexical dependency experts, whose preferences are combined by efficient exact inference. The software can also be used as an unlexicalized stochastic CFG parser. The package comes with models for English, Chinese, German and Arabic; and can be trained for other languages[17]. The parser also provides as output the Stanford Dependencies as well as the phrase structure trees. The Stanford Dependencies are triplets representing grammatical relations between words in a sentence [DMM08]. The *Berkeley Parser* provides Java implementations and PCFGs models for several languages trained using structured probabilistic models, including unsupervised and latent-variable methods [18]. The *Charniak-Johnson Parser* (also known as the Brown Laboratory for Linguistic Information Processing Parser or BLLIP) provides several implementations, in C++, for reranking parsers and maximum entropy parsers along with self-trained parsing models [19]. Many other parsers are available, such as OpenNLP, OpenCCG, Connexor, Collins/MIT parser, MaltParser, among others.

3.1.2 Entity Extraction and Named Entity Recognition (NER)

Entity extraction seeks to identify entities (such as people, organizations or locations) from text, and NER focuses on classifying those entities into predefined categories such as the names of people, organizations, locations, time anchors, quantities, monetary values or percentages, among others. Several hierarchies of named entity types have been proposed. For example, BBN Technologies categories consist of 29 types and 64 subtypes [Bru02]. Another example may be Sekine's extended hierarchy, with 200 subtypes [SSN02][20].

Entity extraction is often performed using linguistic grammar-based techniques like traversing a syntactic tree looking at noun phrase nodes that will be classified later. Concerning NER, handcrafted grammar-based systems typically obtain better precision, but at the cost of lower recall while statistical NER systems typically require a large amount of manually annotated training data.

There are several state-of-the-art NER systems available. The *Stanford NER*, also known as CRFClassifier, provides a general implementation of linear chain conditional random field (CRF) sequence models. Comes with models for up to 7 classes for English text (Time, Location, Organization, Person, Money, Percent, Date) and several other models for German and Chinese can be downloaded. A trainer is included to build sequence models for any arbitrary task [21] [FGM05]. *OpenNLP* includes a rule based and statistical NER. Finally, *GATE* supports NER across many languages and domains.

3.1.3 Coreference Resolution

Coreference resolution is the task of identifying all mentions of entities (and/or events) in a document and determine which of them refer to the same entities. Automatic identification of corefering entities and events has been a difficult problem in NLP, partly because it can require common sense knowledge, and partly owing to the lack of substantial annotated data [PRM+11]. Hand-coded algorithms for coreferences resolution may look for the nearest preceding entity that is compatible with a referring expression. For example, "she" might attach to a preceding expression such as the "woman" or "Alice", but not to "Bill". State of the art techniques use entity-centric, precision-ranked rules learned using machine learning approaches [LPC+11,

[14]Paid: `http://catalog.ldc.upenn.edu/LDC99T42`
[15]Paid: `http://catalog.ldc.upenn.edu/LDC2008T05`
[16]Limited distribution: `http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi`
[17]Available: `http://nlp.stanford.edu/software/lex-parser.shtml`
[18]Available: `https://code.google.com/p/berkeleyparser/`
[19]Available: `https://github.com/BLLIP/bllip-parser`
[20]Available: `http://nlp.cs.nyu.edu/ene/`
[21]Available: `http://nlp.stanford.edu/software/CRF-NER.shtml`

LCP+13]. Moreover, some systems tend to incorporate ad-hoc domain-specific rules and heuristics to solve coreference [GRD10] due to the difficulty of this task.

There are several coreference resolution state-of-the-art systems available such as the *Stanford Deterministic Coreference Resolution System* implements a multi-pass sieve coreference resolution system, [22] or the *Berkeley coreference system* [23].

3.2 Information Extraction and Text Understanding

Information Extraction (IE) focuses on obtaining structured information from given unstructured natural language. Information extraction is relevant for tasks such as document indexing, document retrieval, question answering, text classification, or sentiment analysis.

A common approach to Information Extraction (IE) is to use patterns and regular expressions matched against the text of the document in order to identify items of interest. For example, regular expressions can define character level patterns for identifying named entities in specific domains of bioinformatics (e.g. protein and gene names) [SG06]. In document classification, a common approach called *bag-of-words* is to extract the more salient and relevant words from a document by counting their frequencies and discarding the most common ones. Many sentiment analysis tasks simply look for occurrences of some indexed words within a text, which are then mapped to certain numeric values for different attributes.

Similar to regular expressions, the Stanford NLP group has developed *Tregex* [LA06][24], a collection of utility classes that allow matching patterns in trees, based on tree relationships and regular expression matches on nodes from different annotation layers added by different tasks of the NLP pipeline. Another prominent example of such approach is the *Sundance Parser* and *AutoSlog* systems [RP04]. Sundance uses patterns that match certain tokens and grammatical categories and because those patterns may be complex or not clear, the system is complemented with the AutoSlog utility, intended to be able to extract such patterns from a given set of examples. Combining several approaches, systems like *OpinionFinder* (U. of Pittsburg, Cornell and U. of Utah)[25] provide means of identifying subjectivity and sentiment analysis along various other IE tasks.

Another approach for extracting information from text is to use noun-verb tuples derived from the dependency parse of the document [SG06, YGTH00, CJ09]. These tuples can be further exploited for *text understanding*, which build on top of IE, generally mapping the extracted information to a set of assertions in predicate logic. Text understanding then uses logical inference and deduction to further process the extracted information. The problem of understanding ambiguous words translates to mapping into the proper symbols. Common sense and domain knowledge (Section 3.3) is used to address these issues as well as to provide additional assertions used during the logical deduction process [HSP10].

3.3 Common Sense and Domain Knowledge

Ambiguity is pervasive in natural language. On top of that, the use of neologisms, idioms, non-standard language or the often implicit information or the lack of context make most NLP tasks still very hard even with today's computational power. In order to solve these problems, NLP work often exploits general or domain specific knowledge bases. For example, in some languages noun morphology may determine the gender of certain entity whereas gender can still be identified by knowing that a pregnant mammal is likely to be a female. One application is in word sense disambiguation (WSD), which is critical to the accuracy and reliability of natural language processing [HSP10], since often words with the same spelling can have very different meanings.

Common sense and domain knowledge is not an explicit field in NLP but rather a transversal area of research seeking methods for encoding general common sense or specific domain knowledge so it can be exploited by NLP applications [SM05]. Many representation formalisms have been employed for representing specific domain knowledge, such as atlases, gazetteers or specialized dictionaries for identifying features in a

[22] Available: `http://www-nlp.stanford.edu/downloads/dcoref.shtml`
[23] Available: `http://nlp.cs.berkeley.edu/berkeleycoref.shtml`
[24] Available: `http://nlp.stanford.edu/software/tregex.shtml`
[25] Available: `http://mpqa.cs.pitt.edu/opinionfinder/`

given domain like foreign city names that may be miscategorized as person names because of capitalization heuristics.

Concerning general and common sense knowledge, *WordNet* is a widely used, large-scale database that links English nouns, verbs, adjectives, and adverbs to sets of senses called synsets that are linked to one another through relations of several types like synonymy/antonymy, hypernymy/hyponymy, entailment, etc. WordNet is optimized for lexical categorization and word-similarity determination [Fel99, Mil95]. *ConceptNet* sprung from the Open Mind Common Sense (OMCS) project and is a widely used, large-scale commonsense knowledge base. ConceptNet integrates knowledge from several sources and also provides cross-language links between concepts. ConceptNet is optimized for making practical context-based inferences over real-world texts [LS04]. *Cyc* (commercial), *OpenCyc* (free) and *ResearchCyc* (for research) are large-scale databases of handcrafted commonsense axioms and supporting inference and query engines. Cyc is optimized for formal logical reasoning [Len95].

Some projects exist that focus on verbs and verb frame information. Akin to WordNet, *VerbNet* is a broad-coverage, comprehensive verb lexicon [SM05]. *FrameNet* is a lexicography corpus. It contains descriptions of the semantic frames underlying the meanings of words. Includes the valence representation (semantic and syntactic) of several thousand words and phrases, each accompanied by a representative collection of annotated examples [BFL98]. *PropBank*[26] is an annotated corpus that adds semantic information, more specifically argument structure labels, to the LDC Penn English Treebank.

Other projects focus on real world entities rather than linguistic information. *Freebase* [27], formerly known as Metabase, is a large, collaborative, graph based knowledge source linking topics and facts. Google bought Metaweb and created Knowledge Graph[28], a knowledge base built on top of Freebase used to power the Google search engine. The *Linked Data* project[29] is another attempt at embedding semantic, machine readable information in web pages across the Internet. It relies on standard technologies like the resource description framework (RDF).

On top of the aforementioned projects, there are several projects focused on enhancing or linking between the different knowledge sources. For example *SentiWordNet* [ES06][30] complements WordNet synsets with information for sentiment analysis and the *SemLink* [Pal09] and *Unified Verb Index* link between WordNet synsets and entries in PropBank, VerbNet and FrameNet. Another example may be *NomBank* [MRM+04] an annotation project to complement the PropBank with action nouns for the annotated verb frames. Also, annotated corpora can be obtained in order to train new models or test machine learning algorithms. For example, the *SemCor* corpus[31] is annotated with WordNet synsets.

4 Bridging the Gap

In Section 2 we have seen the importance of narratives, how those can be computationally modeled and how procedural content generation (PCG) can use those models. Then, in Section 3 we looked at using NLP to automatically extract computational models of narratives from natural language text and how common sense and other domain knowledge can be used to enrich those models. This section will present an overview of some systems that combine both fields. Figure 6 shows a high-level overview of how ideas from both of those fields can be integrated into a unified pipeline.

Not much work exists in this area, but the little existing research can be classified in two main groups: those that aim at using NLP to generate computational models of static scene descriptions from text, and those that use NLP to generate computational models of narratives that involve actions, time and change. We describe these two lines of work in the next two sections.

[26] More info: `https://verbs.colorado.edu/propbank/`
[27] Available: `http://www.freebase.com/`
[28] More info: `http://www.google.com/insidesearch/features/search/knowledge.html`
[29] More info: `http://linkeddata.org/`
[30] Available: `http://sentiwordnet.isti.cnr.it/`
[31] Available: `http://www.cse.unt.edu/~rada/downloads.html#semcor`

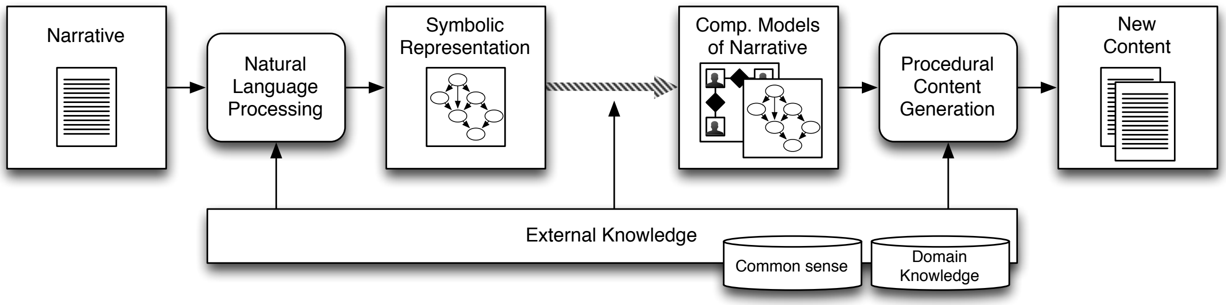

Figure 6: Simplified pipeline for a system exploiting NLP to generate new content from textual narratives. *WordsEye* and *CarSim* feature similar pipelines.

4.1 Extracting Scene Models from Text

Some work focuses on extracting narrative elements, such as characters and their narrative roles from text. For character extraction a common approach used by Calix et. al. [CJKK13] and our own research [VVOZ14] is to extract noun phrases from a syntactic parse tree structure. Then a feature vector is computed for each of the noun phrases and finally a machine learning approach is used to classify the extracted noun phrases as either characters or non-characters. Building on top of this research we look at the "sphere of action" around each character (i.e., the actions that performs or are performed upon) using extracted verb triplets and encoded as "action matrices." Then search the role assignment space and compare with a given reference action matrix in order to find a suitable role assignment for all characters[VVOZ13a].

Elson et. al. focused on extracting other narrative information, namely, character relationships. They identify speech quotes and prosodic cues to attribute speech to named characters. Using the quoted speech attribution they establish relationships based on dialogue between characters [EDM10].

The *WordsEye* system [CS01, CRHS10][32] is a text-to-scene system that creates 3D scenes from natural language descriptions. The system implements an NLP pipeline similar to the one presented in Figure 4 which is combined with a lexical database of common sense knowledge (WordNet, FrameNet and a custom Scenario-Based Lexical Knowledge Resource) to extract a symbolic representation of the scene. The system converts dependency structures into semantic nodes and roles representing spatial relationships and visual attributes. The system relies on a large database of 3D models and poses for entities and actions. The extracted structure is mapped to the database and finally rendered into an image.

4.2 Extracting Narrative Models from Text

Chambers and Jurafsky [CJ09] have used shallow representations and unsupervised learning approaches to automatically learn narrative schemas (i.e., coherent sequences or sets of events) from verb frames. They use unsupervised machine learning to automatically identify narrative event chains by parsing the text, resolving coreference, and extracting chains of events that share participants. The event chains for one character are used for automatically assigning the arguments for the identified verbs. Then verb triplets are built and integrated to form a single narrative schema combining the event chains for each character.

Regneri et. al. [RKRP11] started with the extracted verb frames and event chains and focused on enriching the narrative schemas with information about their participants (e.g., participating characters). They formalized the problem in a similar way to coreference resolution where candidate participant descriptions in event descriptions were identified and later partitioned into sets, each set correspond to specific participants and their members possible mentions to each participant.

The *AESOP* [GRD10] system works on unannotated natural language text and extracts a "plot unit" representation for the narrative. Plot unit structures consist of affect states for each character and links defining the relationships between them. First, they identify characters using a rule-based coreference system and WordNet information. Then, their system uses an automatic process to identify words that correspond to

[32] More info: `http://www.wordseye.com`

positive, negative, and mental affect states. Finally they use the Sundance parser to produce a shallow parse of each sentence and map the affect states onto the characters in the story based on verb frame argument structure.

CarSim [JBDN05] is a system that that automatically converts narratives in the traffic domain into animated 3D scenes. It is intended to be a tool for visualizing traffic situations from text reports in natural language. Unlike WordsEye, CarSim is not limited to recreating static scenes, and considers the time dimension. It also implements an NLP pipeline similar to the one presented in Figure 4 and extracts entities, events, relations and environment attributes separately. Then infers implicit information using a spatio-temporal planning and inference module that produces a full geometric description of the extracted symbolic representation from the text. Time descriptions and the output of the planning module are used to compute trajectories and generate an animation.

5 Synthesis

In this document, in Section 2 we first highlighted the importance narratives in human communication and described the complexity of computational approaches for narrative modeling. We saw applications of computational models of narrative across different disciplines and fields of study. Then, in Section 3 we presented the state of the art of natural language processing (NLP) and how it can be used to automatically extract structured narrative information from text. Finally in Section 4 we described some systems that exploit NLP to extract different kinds of narrative information from text and use it for some specialized cases of procedural content generation (PCG). Figure 6 shows a synthesis of the main ideas covered in this document.

5.1 Open Research Questions

In summary, there are still many open research challenges in the field of computational models of narrative, which we divide in two large groups: those concerning computational narrative, and those concerning natural language processing. Concerning, computational narrative, the key open research questions are:

- How can plot and discourse be modeled computationally? Many approaches exist, such as scripts, plans or grammars but each of those approaches can only capture certain parts of the narrative. Large parts of what constitutes the plot and discourse of narratives, such as authorial intent, conflict, or character subjectivity are not properly captured by existing models.
- How can those models be exploited to analyze and generate narrative? Many automatic or semi-automatic narrative generation systems exist, but their output is still far from the range and quality of narrative generated by human authors.

Concerning natural language processing in the context of extracting narrative information, the key open research questions are:

- Is it possible to automatically extract computational models of narrative from text? Although some approaches exist that can extract with certain accuracy aspects such as actors and their relationships, automatically analyzing text in order to extract plot or discourse models is still an open problem.
- How can narratology theories be exploited to improve the performance of NLP in the context of narrative text? While the general NLP problem might remain unsolved, certain narratology theories, such as Propp's structuralist analysis, provide a framework within which to understand certain type of texts, which might impose additional restrictions, and ease parts of the NLP.
- What is the scalability of NLP techniques and how to reuse NLP models and systems across different domains.

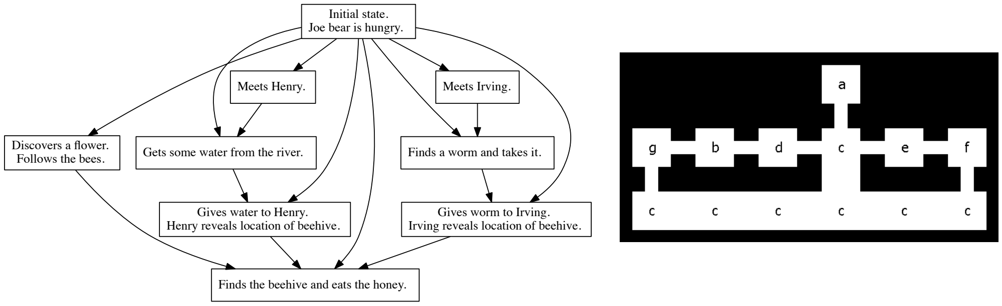

Figure 7: Example story space defined as planning operators and a graphical realization of one of the maps generated by our system.

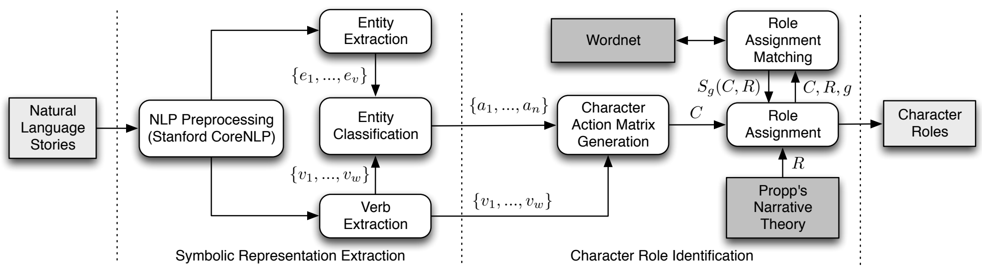

Figure 8: The architecture of *Voz*, our system for extracting characters and identifying their narrative roles from unannotated natural language text.

5.2 Current Work

In order to better understand the state-of-the-art of the current research in the fields covered in this document I have researched both ends of the pipeline described in Section 4 (see Figure 6). From a digital entertainment perspective studied the relationship between computer games and storytelling. I presented a PCG framework to generate game environments (i.e. maps) that supported a subset of stories from a story space defined as planning operators [VVOZ13b]. Figure 7 shows an example of a story space (inspired by Tale-spin's stories) and a graphical realization of one of the maps generated by our system.

Understanding the complexity and the authorial burden of manually generating the models required for such systems I moved to the other end of the pipeline. I have recently worked towards developing a framework that can automatically extract a shallow symbolic representation of the events and existents in a narrative and use such representation combined with domain knowledge to infer higher level narratological information such as the roles the characters play in the narrative [VVOZ13a]. Figure 8 shows the architecture of *Voz*, our system for extracting characters and identifying their narrative roles from unannotated natural language text.

5.3 Future Work

I intend to continue my research computational models of narrative and study the latest developments in natural language processing and knowledge engineering in order to enable the automatic extraction of rich and meaningful structures from unannotated text that can be used in a procedural content generation application. I would like to be able to connect my latest research on extracting narratological information from text with my initial work on a game map generation in order to automatically generate game maps based on a given story in natural language. In order to achieve this goal I need to further research approaches to natural language understanding that allow me to bridge the gap between the shallow representation that can be extracted from a text and the higher level structures that are inferred from a narrative. I plan on incorporating common sense and domain knowledge using approaches inspired by cognitive science.

References

[All84] James F. Allen. Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154, 1984.

[Bat97] John A. Bateman. Enabling technology for multilingual natural language generation: the kpml development environment. *Nat. Lang. Eng.*, 3(1):15–55, March 1997.

[BFL98] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, 1998.

[Bob64] Daniel G. Bobrow. *Natural language input for a computer problem solving system*. PhD thesis, Massachusets Institute of Technology, 1964.

[Bru02] Ada Brunstein. Annotation guidelines for answer types. *BBN technologies report*, 2002.

[Cha80] Seymour Benjamin Chatman. *Story and discourse: Narrative structure in fiction and film*. Cornell University Press, 1980.

[CJ09] Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, pages 602—-610, 2009.

[CJKK13] Ricardo A. Calix, Leili Javadpour, Mehdi Khazaeli, and Gerald M. Knapp. Automatic Detection of Nominal Entities in Speech for Enriched Content Search. *The Twenty-Sixth International FLAIRS Conference*, pages 190–195, 2013.

[CL96] Jim Cowie and Wendy Lehnert. Information extraction. *Communications of the ACM*, 39(1):80–91, 1996.

[CRHS10] Bob Coyne, Owen Rambow, Julia Hirschberg, and Richard Sproat. Frame semantics in text-to-scene generation. In *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 375–384. Springer, 2010.

[CS01] Bob Coyne and Richard Sproat. Wordseye: an automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 487–496. ACM, 2001.

[DMM08] Marie-Catherine De Marneffe and Christopher D Manning. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics, 2008.

[DP11] Jonathon Doran and Ian Parberry. A prototype quest generator based on a structural analysis of quests from four MMORPGs. *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games - PCGames '11*, pages 1–8, 2011.

[EDM10] David K Elson, Nicholas Dames, and Kathleen R McKeown. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147. Association for Computational Linguistics, 2010.

[EM07] David K. Elson and Kathleen R. McKeown. A platform for symbolically encoding human narratives. *Proceedings of the AAAI Fall Symposium on Intelligent Narrative Technologies*, pages 29–36, 2007.

[ES06] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422, 2006.

[Fai07] Chris R. Fairclough. *Story Games and the OPIATE System.* PhD thesis, University of Dublin, Trinity College, 2007.

[FC03] Chris R. Fairclough and Pádraig Cunningham. A multiplayer case based story engine. *Computer Science Technical Report*, 2003.

[Fel99] Christiane Fellbaum. *WordNet.* Wiley Online Library, 1999.

[FGM05] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[Fin09] Mark Alan Finlayson. Deriving narrative morphologies via analogical story merging. In *Proceedings, 2nd International Conference on Analogy*, pages 127–136, 2009.

[Fin11] Mark A. Finlayson. The Story Workbench: An Extensible Semi-Automatic Text Annotation Tool. *Intelligent Narrative Technologies*, pages 21–24, 2011.

[Flu96] Monika Fludernik. *Towards a 'natural' narratology.* Routledge London, 1996.

[GDAPH05] Pablo Gervás, Belén Díaz-Agudo, Federico Peinado, and Raquel Hervás. Story Plot Generation based on CBR. *Knowledge-Based Systems*, 2005.

[Ger09] Pablo Gervás. Computational Approaches to Storytelling and Creativity. *AI Magazine*, pages 49–62, 2009.

[GL83] Gérard Genette and Jane E. Lewin. *Narrative discourse: An essay in method.* Cornell University Press, 1983.

[GRD10] Amit Goyal, Ellen Riloff, and Hal Daumé, III. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 77–86, Stroudsburg, PA, USA, 2010.

[Her00] David Herman. Narratology as a cognitive science. *Image and Narrative*, 1:1, 2000.

[HPI10] Hugo Hernault, H. Prendinger, and M. Ishizuka. HILDA: a discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3):1–33, 2010.

[HSP10] Catherine Havasi, Robert Speer, and James Pustejovsky. Coarse word-sense disambiguation using common sense. In *AAAI Fall symposium series*, 2010.

[HZDR11] Ken Hartsook, Alexander Zook, Sauvik Das, and Mark O. Riedl. Toward supporting stories with procedurally generated game worlds. *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, pages 297–304, August 2011.

[JBDN05] Richard Johansson, Anders Berglund, Magnus Danielsson, and Pierre Nugues. Automatic text-to-scene conversion in the traffic accident domain. In *IJCAI*, volume 5, pages 1073–1078, 2005.

[KM03] Dan Klein and CD Manning. Accurate unlexicalized parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423—-430, 2003.

[LA06] Roger Levy and Galen Andrew. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on Language Resources and Evaluation*, pages 2231–2234. Citeseer, 2006.

[Lab72] William Labov. *Language in the Inner City: Studies in the Black English Vernacular.* University of Pennsylvania Press, 1972.

[LCP+13] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. 2013.

[Len95] Douglas B. Lenat. CYC : A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[LPC+11] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics, 2011.

[LS04] H. Liu and P. Singh. ConceptNet a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

[Mal01] Scott A. Malec. Proppian structural analysis and xml modeling. *Proceedings of Computers, Literature and Philology (CLiP 2001)*, 2001.

[Man12] Inderjeet Mani. *Computational Modeling of Narrative*, volume 5 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers, 2012.

[Mee76] James Meehan. *The Metanovel: Writing Stories by Computer*. Ph.d., Yale University, 1976.

[Mil95] George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[MJ12] Christopher Manning and Dan Jurafsky. nlp-class.com (mooc), 2012.

[MMS93] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[MP04] Inderjeet Mani and James Pustejovsky. Temporal discourse models for narrative structure. In *Proceedings of the 2004 ACL Workshop on Discourse Annotation*, pages 57—-64, 2004.

[MRM+04] A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. The nombank project: An interim report. In A. Meyers, editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

[MS99a] Christopher D. Manning and Hinrich Schuetze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[MS99b] Michael Mateas and Phoebe Sengers. Narrative intelligence. In *Proceedings AAAI Fall Symposium on Narrative Intelligence*, pages 1–10, 1999.

[MT88] William C. Mann and Sandra A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[Pal09] Martha. Palmer. SemLink: Linking PropBank, VerbNet and FrameNet. In *Proceedings of the Generative Lexicon ConferenceGenLex-09*, Pisa, Italy, Sept 2009.

[PCI+03] James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34, 2003.

[PDL+08] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse Treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, 2008.

[PRM+11] Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27. Association for Computational Linguistics, 2011.

[Pro73] Vladimir Propp. *Morphology of the Folktale*. University of Texas Press, 1973.

[RB12] Alejandro Ramirez and Vadim Bulitko. Telling Interactive Player-Specific Stories and Planning for It: ASD+ PaSSAGE= PAST. *Eighth Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2012)*, pages 173–178, 2012.

[RFW09] Whitman Richards, Mark Alan Finlayson, and Patrick Henry Winston. Advancing computational models of narrative. Technical report, CSAIL Technical Report, 2009.

[RKRP11] Michaela Regneri, Alexander Koller, J Ruppenhofer, and Manfred Pinkal. Learning Script Participants from Unlabeled Data. *RANLP*, 2011.

[RP04] Ellen Riloff and William Phillips. An introduction to the sundance and autoslog systems. Technical report, Technical Report UUCS-04-015, School of Computing, University of Utah, 2004.

[RSDA08] Mark O Riedl, Andrew Stern, Don Dini, and Jason Alderman. Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, 4(2):23–42, 2008.

[RTB11] Mark Riedl, David Thue, and Vadim Bulitko. Game ai as storytelling. In *Artificial Intelligence for Computer Games*, pages 125–150. Springer, 2011.

[Rya87] Marie-Laure Ryan. On the window structure of narrative discourse. *Semiotica*, 64(1-2):59–82, 1987.

[SA77] Roger C. Schank and Robert Abelson. Scripts, goals, plans, and understanding, 1977.

[SA13] Roger C. Schank and Robert P. Abelson. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press, 2013.

[SG06] Mark Stevenson and Mark A. Greenwood. Comparing Information Extraction Pattern Models. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 12–19, 2006.

[SLTW05] Lewis Seifert, Celeste Lim, Laura Tan, and Nicole Wee. Digital propp: Proppian fairy tale generator, 2005.

[SM05] Lei Shi and Rada Mihalcea. Putting pieces together: Combining FrameNet, VerbNet and WordNet for robust semantic parsing. In *Computational Linguistics and Intelligent Text Processing*, pages 100–111. Springer Berlin Heidelberg, 2005.

[SPK06] Robert Scholes, James Phelan, and Robert Kellogg. *The Nature of Narrative: Revised and Expanded*. Oxford University Press, 2006.

[SSN02] Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. Extended named entity hierarchy. In *LREC*, 2002.

[Tur93] Scott R. Turner. Minstrel: a computer model of creativity and storytelling. 1993.

[TYSB10] Julian Togelius, Georgios Yannakakis, Kenneth Stanley, and Cameron Browne. Search-based procedural content generation. *Applications of Evolutionary Computation*, pages 141–150, 2010.

[VVOn12] Josep Valls-Vargas and Santi Ontañón. Natural Language Generation through Case-Based Text Modification. *Case-Based Reasoning Research and Development*, pages 443–457, 2012.

[VVOZ13a] Josep Valls-Vargas, Santiago Ontanón, and Jichen Zhu. Toward character role assignment for natural language stories. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013.

[VVOZ13b] Josep Valls-Vargas, Santiago Ontanón, and Jichen Zhu. Towards story-based content generation: From plot-points to maps. In *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, pages 1–8. IEEE, 2013.

[VVOZ14] Josep Valls-Vargas, Santiago Ontanón, and Jichen Zhu. Toward automatic character identification in unannotated narrative text. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.

[YGTH00] Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 940–946. Association for Computational Linguistics, 2000.